Semantics based analysis of botnet activity from heterogeneous data sources

Santiago Ruano Rincon, Sandrine Vaton, Antoine Beugnard, Serge Garlatti

Semantics based analysis of botnet activity from heterogeneous data sources

Santiago Ruano Rincón, Sandrine Vaton, Antoine Beugnard, Serge Garlatti
Institut Mines-Télécom
Télécom Bretagne
29238 Brest Cedex 3 - France
Email:{santiago.ruano-rincon,sandrine.vaton,antoine.beugnard,serge.garlatti}@telecom-bretagne.eu

***Abstract*—The diversity in network devices, protocols, data sources and probes impose different challenges to uniformly measure and analyse network traffic. Analysing a network means considering distinctive reporting approaches and diverse methods to represent data, measure times or identify nodes. In this work, we tackle these challenges by relying on semantics, taking advantage of the ontologies' ability to map high-level network concepts to concrete information sources of different nature. In particular, we propose a simple architecture to map network concepts to data stored in relational databases. Based on this architecture, we implement a tool that looks for malicious bot activity, studying, from a unique point of view, DNS traffic from PCAP sources, and TCP connections from IPFIX reports. This approach is able to enhance current DNS based botnet detection methods, taking into account additional heterogeneous analysis elements.**

I. Introduction

Networks' capability to interconnect different devices, protocols, management systems and information sources necessarily yields to heterogeneous environments. This asset challenges network managers, for whom it is difficult to holistically manage their systems, having to make use of large, isolated and diverse measurement solutions. As we describe in the following section, different authors have identified several issues related to heterogeneity, and, according to their works, we hypothesise here that semantics (the study of the relationship of meanings of a sign) is able to solve those issues, correlating the different information sources through a high conceptual level.

One of most the relevant issues in network traffic analysis, is to identify botnets. More than any other source of network anomalies, botnets represent the most significant medium to carry out malicious activities today, such as denying services, spamming, phishing and extorting business data [4], [6]. This network management need has motivated us to develop a holistic approach, and to evaluate its capacity to identify botnets.

The central element of our approach are ontologies, that is to say formal knowledge representations. Ontologies can provide a semantic layer between concrete data and network concepts. In other words, they make it possible for the network manager to work with instances of concepts (such as latency measures, a web server address, and timestamps), instead of raw data.

Under these statements, we make use of ontologies to evaluate the presence of bots in the local network. In this case, we study two different data sources, i.e., DNS traffic from PCAP captures, and the TCP level information from IPFIX reports. As we explain in the following section, DNS provides a sound foundation to state-of-the-art bot detection approaches Indeed, it makes it possible to find traces of botnet collective behaviour, such as the look up for the IP addresses of the rendezvous point or the victims. At the same time, we can identify SYN flooding attacks in the TCP information provided by IPFIX.

The main contribution of our work is, thanks to ontologies, it permits to relate different datasets to each other, even if they have been collected in distinct raw formats by different probes. Ontologies provide a common language to give meaning to the different data and then put them into the same network analysis context. In this case study, we take advantage of this semantic asset to identify botnet activity.

In the rest of this paper we first present related works. Then, we present how we have built the ontology considering state-of-the-art methods to detect botnets. We also describe the architecture that provides a unique access point to query different network management elements. Finally, we show the results of the scenario that we have implemented, analysing a dataset from a university computer room, reporting possible bots which some of them are certainly carrying out SYN flooding.

II. Related works

The related research of this work is twofold. From a general perspective, it addresses challenges resulting from network heterogeneity. From a specific point of view, we evaluate the ability of semantic tools to detect botnet activity. In this section we study significant work regarding both issues.

A. Related work on network heterogeneity

Different researchers have characterised issues associated to network heterogeneity. For example, Wong et al. [18] describe interoperability problems in router management. Also, López et al. [10] identify, at least, three different types of issues concerning measurement: how to name devices and components, how to represent data and how to measure units.

While in this work we focus on semantic solutions, there are other techniques that address this problem. For example, Zurawski et al. [20] and Hanemann et al. [8] have proposed solutions based on structured languages, specifically on XML

Schema, able to translate data information from different sources. However, these approaches are limited to the static representation they provide.

Semantic based solutions to solve interoperability or heterogeneity issues have been considered by several authors, such as Ferreiro et al. [7], López et al. [10], Wong et al. [18] and XuHui et al. [19].

Wong et al. [18] state that ontologies bring some benefits to network measurement, such as interoperability of information models and high-level design expressiveness. The authors formulate an ontology set addressing the interoperability challenges to configure network routers, focusing on Cisco and Nortel.

The most prevailing work concerning measurement approaches is the Measurement Ontology for IP traffic (MOI), that aims to semantically interface management systems and measurement devices [13]. MOI is based on the works by Ferreiro et al. [7], López et al. [10] and the MOMENT [17] project, and it is currently under standardisation at ETSI Industrial Standardization Group (ISG).

MOI design is inspired by the network measurement needs of five use cases: network characterization, QoS measurement, traffic monitoring for security applications, autonomic network monitoring and management, and law enforcement [12].

MOI considers network *measurements* as its central ontology concept, embracing data related to measures such as ping, traceroute or the identity of network objects. MOI aims to structure the network concepts in different modules: IP traffic measurement ontologies, IP network monitoring systems, parameters of applications' quality, QoS-QoE correlation mechanisms, quality control systems, and transversal security and privacy. However, for the moment, the most recent document describing the implementation of MOI focuses on lower layers of the ontology. The current MOI architecture consists of five ontologies: General Concepts, Units, Metadata, Data and Anonymity.

In the MOMENT project, a relevant MOI's base, López et al. [10], [17] formulate an approach focused on the above ontologies to measure heterogeneous network data sources. The MOMENT project proposes a comprehensive architecture able to integrate a wide range of probes, measurement infrastructures and analysis services.

We can conclude that MOI represents the most important work on network measurement ontologies, and we aim to integrate their existing concepts in the scope of this work.

Additionally, an essential component of the MOMENT proposal is the Relational Database (RDB) to Resource Description Framework (RDF) semantic mediation, for which we can find different tools [16] such as D2RQ, Virtuoso RDF View and R2O, none of them prevailing over the other. However, in this work we can consider D2RQ since it is open source, stable and used in MOMENT [10].

B. Related work on detecting botnet activity

The use case considered in this paper concerns the detection of botnets. The term botnet packs two words together that explain its nature: robot and network. It is indeed a network that, in its illegal version, is composed of hosts compromised by a malicious software under the control of a bot master [4]. In this paper we focus on botnet detection methods that rely on the analysis of Domain Name System (DNS) traffic.

To detect botnets is a difficult task, mainly because botnets highly differ from each other and evolve over time. Despite their particularities, bots carry out three common tasks: infect new devices, look for instructions in a rendezvous point, and execute their main function, such as carrying out a Distributed Denial of Service (DDoS) or sending spam. Among these three tasks, we can state, from the work by Feily et al. [6], that the search for the rendezvous point is the most accurate and general analysis object to detect malicious activity.

Feily et al. also categorize different detection methods according to four main focus: botnet signatures, network anomalies, DNS traffic, and data mining [6]. They conclude that the two most relevant and promising types of methods rely on the analysis of DNS traffic or data mining. In this work, we focus on DNS based detection methods, since as we can conclude from [6], they may be lightweight, and depending on the approach, able to detect a large number of botnets, even if they perform encrypted communications.

DNS is an essential Internet protocol. It is used by any service requiring to translate, for example, human-readable and easy-to-remember domain names into machine-manipulable Internet Protocol (IP) addresses. The electronic mail service also depends on DNS service to look for the mail exchangers (MX) for a certain domain. In a similar way, sophisticated botnets require DNS to determine the IP addresses of the rendezvous point and of their victims, or in the case of spamming bots, to search for mail servers to send spam through.

Botnets and DNS: According to Bianchi et al. [2] and Choi et al. [4], botnets' behaviour is observable in DNS traffic in different aspects. For example, a high ratio of requests for non-existing domains (NXDomain) over existing domains (NoError) likely means the search of a hidden rendezvous point's IP address; an excessive number of simultaneous lookups for a single domain from different sources possibly denotes bots searching the victim's address, preparing a coordinated attack; domain name-to-IP address records changing too fast and zones with too short Time to Live (TTL) could indicate methods to protect the rendezvous point. These metrics are considered by the botnet detection methods from the DEMONS project and by Choi et al. [4] that we describe here.

DEMONS botnets detection methods: The European project DEMONS considers two different botnet detectors: a general botnet detector and StreaMon [2], that focuses on Conficker.

Conficker has been one of the most spreading botnets in the last years. It takes advantage of Microsoft's File sharing service of vulnerable hosts (445/TCP port) to spread over the network [9]. As for any botnet, infected hosts need to contact a Command and Control (C&C) server to look for instructions. Conficker hides the rendezvous point with their C&C in an obfuscated domain name, that the bots try to find performing an extremely high number of DNS queries, and then, increasing the probability to get NXDomain (non-existing domain) answers. Streamon thus searches for bots analysing

per host the number of DNS queries, the number of NXDomain answers and the number of TCP SYNs and SYN/ACKs to and from port 445. It is worth to note that Streamon measures these data from PCAP sources, while we aim to integrate additional heterogeneous probes, e.g. IPFIX reports.

On the other hand, the DEMONS general botnet detector relies on two different analysis regarding the two possible answers to DNS queries, NXDomain or NoError. The NXDomain based analysis focuses on detecting "domain lux" botnets, which frequently change the domain name of their rendezvous point, to escape from classical blacklisting. Examples of these botnets are Conficker, Kraken/Bobax, Srizbi and Torpig. In contrast, the NoError based approach focuses on detecting malicious domain names, focusing on the unusually fast-changing domain name-to-IP address mappings. This approach inspects DNS messages to classify domain names, according to different metrics, such as: the number of queries for the domain, the Time-To-Live of A-Records, and the list of IP addresses to which the domain was mapping.

BotGAD (Botnet Group Activity Dectector): Choi et al.[4] have developed "BotGAD", a lightweight and robust method relying on Domain Name Service (DNS) traffic. Choi et al. [4] characterise the botnets' "group activity" or collective behavior, such as coordinated and simultaneous attacks, that BotGAD detects through non-supervised machine learning. Contrary to the DEMONS' method, BotGAD only analyses DNS traffic.

BotGAD builds matrices that relate domain names to timestamps and client IPs of DNS queries for such domains, and then compare similarities among domain matrices to detect new botnet activity. Choi et al. state that BotGAD is able to detect botnets that follow recent evasion methods, in large-scale networks, in real-time, and even if they encrypt their messages.

From these works, we can conclude that botnet detection methods find in DNS a comprehensive analysis base. Also we conclude that, in the scope of this work, at a first stage, we can focus on the NXDomain/NoError answer ratio to evaluate if each host is acting as a bot, searching for its rendezvous point.

III. A SEMANTIC APPROACH TO ANALYSE BOTNET ACTIVITY THROUGH HETEROGENEOUS DATA

As stated above, in a general perspective, our work addresses issues related to heterogeneity, and it aims to develop an holistic management network support able to uniformly analyse network traffic from heterogeneous data sources. In order to evaluate this approach we focus on one important issue faced by network managers: to detect malicious botnet activity in the local network. As Figure 1 illustrates, this botnet analysis focuses on two heterogeneous data sources, DNS traffic and IPFIX flow reports, that we access through ontologies, creating a single point of view. In this section we first describe the rationale behind our approach and the method that allows to build the ontologies. Then, we illustrate the architecture we have implemented and the results that make it possible to conclude that jointly analysing DNS data and IPFIX flows, from a single point, enhances the evaluation of botnet activity.

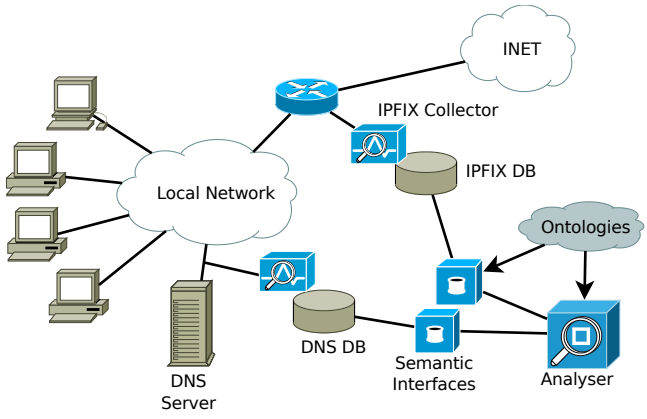

Fig. 1. Botnet detection scenario. The analyser accesses DNS data from PCAP captures and TCP information from IPFIX as RDF graphs, thanks to a semantic interface.

A. Rationale

To analyse more holistically the possible activity of botnets, we rely on current DNS botnet detection methods, and we enhance their outcome by evaluating diverging sources of information. Our approach looks for two bot-related actions, which are usually analysed separately in current solutions. First, as we have explained in Section II, sophisticated bots search their rendezvous point in domain names according to previously specified obfuscated patterns. State-of-the-art detection methods are able to find traces of this rendezvous search in the DNS lookup traffic. Second, a common goal of bots is to carry out DDoS attacks, through methods such as TCP SYN flooding. We can find evidence of this kind of attack in the TCP information from IPFIX reports.

As we have stated in Section II, the analysis of the NXDomain/NoError ratio, from the DNS lookup answers, represents an accurate and simple metrics to evaluate if a host is acting as a bot. We can thus rely on a simplified version of the method proposed in the framework of the DEMONS project and by Bianchi et al. [2], that we illustrate in Algorithm 1.

Algorithm 1 Base algorithm to evaluate bot activity from each host

$NoErrorPerHost \leftarrow$ Select the number of NoError answers per host
$NXDomainPerHost \leftarrow$ Select the number of NXDomain answers per host
for i in $NoErrorPerHost$ **do**
  $Ratio[i] \leftarrow NXDomainPerHost[i]/NoErrorPerHost[i]$
  if $Ratio[i] > Threshold$ **then**
    AddPossibleBots(i, TimeStamp)
  end if
end for

On the other hand, to execute a SYN flood attack, a bot floods its victims interrupting the three-way handshake. Hence, we need to look for TCP connection requests (SYN flag only) with no subsequent packets in the same flow direction (ACK flag only). As we describe below in this section, IPFIX private Information Elements (IEs) [3] include TCP information in their reports, that we can study as shown in Algorithm 2.

Algorithm 2 Base algorithm to find hosts involved in TCP-SYN flooding

```
TcpConnectionFlags ← Select the number of interrupted TCP connections per source host
for i in GETPOSSIBLEBOTS do
    if TcpConnectionFlags > Threshold then
        ADDATTACKINGBOTS(i, TimeStamp)
    end if
end for
```

By means of ontologies, RDF and SPARQL standards, it is possible to link the interpreted data from different distributed sources. Thus it allows us to relate to the same network node and measurement data to the different information sources. Since the base analysis are stored in relational databases, we need to rely on tools, such as D2RQ, that map their data to semantic descriptions, represented in RDF graphs. Later in this section we give an overview of the SPARQL queries that we rely on in this use case.

B. Building the ontology to detect botnets

To design and build ontologies that conceptualise the network measurement data, we have followed these steps, relying on the works by [14], [1]: define the purpose, scope and requirements; model the concepts, their properties and relations; represent the conceptual model in a formalism readable by machines; and evaluate the ontology.

Purpose, scope and requirements: We need an ontology that conceptualises aspects related to DNS, IPFIX flow reports and TCP connections. Also, given that our approach will look for information in databases, the ontology must map the fields from the databases to the ontological representation (RDF graphs) of these concepts. It is important to note that to consider other network-related concepts we rely on the MOI ontologies, taking advantage of the extensibility of ontologies.

Concepts, properties and relations: First, in the general scope of DNS, each message may be composed of five sections [11]: *Header*, *Question*, *Answer*, *Authority (Name servers)* and *Additional (information)*. In this work, we need to consider the components of messages related to *lookups*, since we have chosen to focus on DNS botnet detection methods that rely on them. The analysis requires *Response* information, such as the result (NoError or NXDomain Error), the time of the query, and the addresses of client and server. This information is found in the *Header* section, which embraces the following fields: *Query ID*, *Question/Response flag*, *Response code*, *Time*, *Timestamp*, *Source address* and *Destination address*. We need to store the information for each DNS message header and create an equivalent concept in the ontology.

Figure 2 shows a summarised version of the DNS ontology we have designed, including the linked MOI concepts. *DNSData* is a child of *MeasurementData*, a MOI's high-level concept that models any information carried by any measurement. All the DNS data composing the message header, described above, are subclasses of *DNSData*.

Second, to evaluate whether TCP SYN-flooding attacks are happening in the network or not, we need to handle concepts concerning IPFIX reports and the TCP-related information

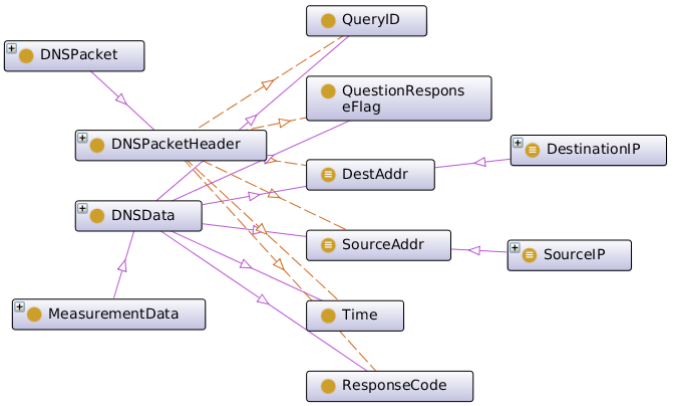

Fig. 2. Main concepts composing the DNS ontology

they provide. In these reports, we search for incomplete TCP connections, that is to say, TCP flows from the originating host with only initial SYN flagged packet and missing ACK messages. IPFIX makes it possible thanks to two variables available in the private Information Elements (IEs) that supports TCP: *initialTCPFlags*, "the TCP flags on the first TCP packet in the flow" and *unionTCPFlags* "the union of the TCP flags on all packets after the first TCP packet in the flow [3]."

IPFIX reports also include a general set of data that covers: *Source* and *Destination addresses* (IPv4 and IPv6), *Source* and *Destination ports*, *Start* and *End times*, and *Total packet count*.

Figure 3 abstracts relevant concepts from the IPFIX ontology we have designed, that we require to detect botnets in the presented approach. Similarly to the DNS ontology, the information from IPFIX reports are modeled under the MOI measurement data class.

It is important to note that we aim to semantically map the data from the sources involved in the analysis. In this case, it is possible to link how the datasets identify the network nodes through the *SourceIP* and *DestinationIP* concepts from the MOI ontologies [13], (See Figure 2 and 3). As well, the *Timestamp* concept relates the time information in DNS messages and flow reports.

Table I lists the DNS and IPFIX required concepts and the existing MOI equivalents that we link in this analysis.

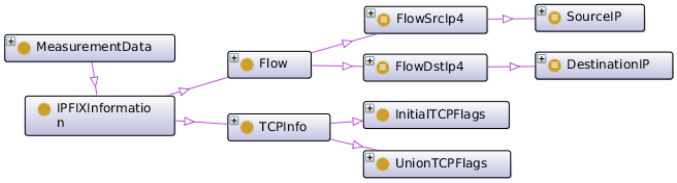

Fig. 3. Part of the IPFIX ontology and TCP flags-related concepts

Represent the formalism in a machine-readable language: We aim to access the network data semantically represented in the RDF model. This model is based on subject-predicate-object triple patterns, also called triples. Since in this use case the data are stored in relational databases, we rely on the D2RQ mapping tool, and we need then to

TABLE I. REQUIRED CONCEPTS AND MOI EXISTING EQUIVALENTS

DNS Concept	TCP/IPFIX	MOI current equivalent
Source address	Source address	SourceIP
Destination address	Destination address	DestinationIP
Question/Response flag		
Response code		
Timestamp	Timestamp	TimeStamp
	Source port	SourcePort
	Destination port	DestinationPort
	Start time	
	End time	
	Total packet count	
	Initial TCP flags	
	Union TCP flags	

describe the ontology in the Turtle based D2RQ mapping language. Similarly to SPARQL, Turtle formats the data in subject-predicate-object RDF triples. In this scenario, we need two different D2RQ ontologies to map the DNS and IPFIX databases.

C. Architecture

Three main layers compose our approach: a probe-and-storage, a semantic interface and the analyser. This architecture relies on a proposal by López de Vergara et al. [10], which focuses on semantic interfaces to relational databases.

Different components in the probe-and-storage layer gather data from the network and store relevant information in relational databases. In this botnet detection use case, this layer is made of a PCAP probe (to filter and collect DNS traffic) and an IPFIX collector (to store TCP traffic reports). We have implemented the PCAP/DNS probe adapting the Blockmon monitoring tool [5]. As IPFIX probe, we have used the YAF tool suite, particularly the YAF MySQL Mediator, that collects IPFIX reports and insert them into a relational database.

The semantic interface layer serves a SPARQLendpoint for each relational database, making it possible to access it as RDF graphs. This layer depends on the ontologies that map the concepts to database table and columns. In this case, we need two mapping ontologies characterizing DNS and IPFIX concepts.

The analyser layer queries the semantic interfaces using SPARQL to retrieve the required information. It is in this layer that we can measure the network and analyse botnet activity. In this approach, to measure the NXDomain/NoError ratio we rely on two SPARQL queries. For example, the following query asks the semantic interface to count the number of non-existing domain answers per host. The query restricts the data to DNS message headers (subject) whose QR_flag (predicate) means they are answers (object); and whose Response code (p) equals 11 in binary, signifying NXDomain error (o). See Lines 6 and 7:

```
1 SELECT (COUNT(DISTINCT ?dnsmessage) as ?no)
2   ?dest_addr
3 WHERE {
4   ?dnsmessage a holmondns:DNSMessageHeader ;
5     MD:DestinationIP ?dest_addr ;
6     holmondns:DNSMessageHeader_QR_flag "1" ;
7     holmondns:DNSMessageHeader_R_code "11" .
8 }
9 GROUP BY ?dest_addr
```

Similarly, we also query the semantic interface to detect SYN flooding attacks. The following query corresponds to the first sentence in Algorithm 2. This SPARQL query is restricted to triples of flows (subject) whose first TCP packet are flagged (predicate) SYN only (object), and the union of the TCP flags on all subsequent packets in the flow (predicate) is empty (object). See Lines 6 and 7 in the query.

```
1 SELECT (COUNT(DISTINCT ?id) as ?no) ?srcip
2 WHERE {
3   ?flowmsg holmonflows:flows_id ?id ;
4     MD:SourceIP ?srcip ;
5   ?tcpmsg holmonflows:tcp_id ?id ;
6     holmonflows:tcp_initialTCPFlags "S" ;
7     holmonflows:tcp_unionTCPFlags "" .
8 }
9 GROUP BY ?srcip
```

Note in these queries that we are integrating the MOI as external ontologies. In these examples, the *DestinationIP* and *SourceIP* concepts, identified by the *MD:* prefix.

The tools and ontologies that we have used and developed to implement this architecture are available online at http://perso.telecom-bretagne.eu/santiagoruano-rincon/holmon/ , including instructions on how to reproduce a similar case use.

D. Validating the approach

To validate our approach we have processed a set of captures from a 19-host students computer room of the Università di Roma Tor Vergata, where it was known that several hosts were infected by malware. The capture set comprises 1 hour and 14 minutes of traffic, that we have probed with the PCAP and IPFIX tools described above.

Table II summarises the offline analysis of such data. Considering a NXDomain/NoError ratio threshold > 0.10 as abnormal behaviour, the analysis hypothesises that the top ten hosts are possible bots. However, the hypothesis is stronger for the top four: not only their NXDomain/NoError ratio is higher than 0.79, but also the number of their TCP connection requests with no subsequent packets are, at least, twice the average (4982.46). This leads to conclude on SYN floods sourcing from their IP addresses.

Regarding performance requirements, a four-processors 2.67GHz CPU running Linux analysed the DNS-related data in 4.235 seconds, in average. However, D2RQ's optimizing option (–fast) has allowed to reduce the required time to 0.274 seconds, though increasing instability risks, according to D2RQ documentation.

We can also compare our approach to the methods described in the Related Works section. Streamon is able to analyse two different types of data: DNS and 445/TCP port related traffic. However, Streamon inspects only raw traffic from PCAP captures. On the other hand, BotGAD [4] performs a highly accurate report and it is able to evolve over time, learning new collective bot behaviour. However, it relies only on DNS traffic.

IV. CONCLUSIONS AND FUTURE WORK

Ontologies make it possible to analyse the status of a network through a unified data model. In this work, we have relied

TABLE II. SUMMARISED RAPPORT OF BOTNET ANALYSIS

#	Host	NXDomain/NoError	Aborted TCP conn.	Bot probability
1	host.139	0.93	18240	SYN Flooding
2	host.142	0.92	14725	SYN Flooding
3	host.147	0.87	27206	SYN Flooding
4	host.146	0.79	13865	SYN Flooding
5	host.71	0.25	0	Suspicious
6	host.63	0.25	0	Suspicious
7	host.62	0.25	0	Suspicious
8	host.97	0.23	123	Suspicious
9	host.88	0.13	110	Suspicious
10	host.144	0.12	0	Suspicious
11	host.87	0.09	131	Unknown
12	host.92	0.06	100	Unknown
13	host.75	0.06	0	Unknown
14	host.94	0.04	107	Unknown
15	host.96	0.03	130	Unknown
16	host.53	0	0	Unknown
17	host.65	0	0	Unknown
18	host.82	0	133	Unknown
19	host.83	0	113	Unknown
		Threshold: 0.10	Avg: 4982.46	

on ontologies and semantic tools to evaluate the botnet activity in a network by uniformly analysing two heterogeneous data sources. We have designed a three-layered architecture: probe-and-storage, semantic interface and analyser layers, and implemented it through a tool that helps to evaluate whether local hosts act as malicious bots or not. Specifically, the tool also helps to assess if malicious bots are SYN flooding their victims.

To carry out this analysis, we have considered two data sources: DNS messages captured by a PCAP probe, and TCP information included in IPFIX reports. Our implemented tool analyses both data sources as RDF data, querying two SPARQLendpoints, which are themselves served by semantic interfaces against data stored in relational databases. We have validated this case study with an actual data set, which allows us to conclude that semantics provide an interesting foundation to holistically analyse networks, thanks to its capacity to handle information at a conceptual level.

Moreover, semantic tools present additional advantages that can further improve a holistic analysis. For example, they make it possible to share knowledge with different or external agents. In this scope, current SPARQL specification [15] allows for running queries that directly link data from different sources such as the DNS and IPFIX endpoints. However, the D2RQ available today lacks support for federated queries.

Semantics also makes it possible to infer knowledge regarding the status of the network. In future work, we could take advantage of semantic reasoners to analyse the probability that a host is acting as a bot, or to detect abnormal behaviour.

In future work we would like to explore these features, and to expand our architecture by considering other data sources, in order to have an even more holistic view of the network, simplifying and unifying network analysis.

V. ACKNOWLEDGMENTS

We would like to thank Professor Giuseppe Bianchi of Università di Roma Tor Vergata, who has kindly provided us with the captures containing known botnet traffic.

REFERENCES

[1] M. P. Adaa, "Ontology for Host-based Anomaly Detection," Master's thesis, University of Oslo, 2007.

[2] G. Bianchi, M. Bonola, G. Picierro, S. Pontarelli, and M. Monaci, "StreaMon: A software-defined monitoring platform," in *Teletraffic Congress (ITC), 2014 26th International*, Sept 2014, p. 1–6.

[3] E. Boschi, B. Trammell, L. Mark, and T. Zseby, "Exporting Type Information for IP Flow Information Export (IPFIX) Information Elements," RFC 5610, Internet Engineering Task Force, July 2009.

[4] H. Choi and H. Lee, "Identifying botnets by capturing group activities in DNS traffic," *Comput. Netw.*, vol. 56, no. 1, p. 20–33, Jan. 2012.

[5] A. Di Pietro, F. Huici, N. Bonelli, P. Kastovsky, S. Vaton, M. Dusi, T. Groleat, and B. Trammell, "Toward Composable Network Traffic Measurement," in *INFOCOM 2013 : 32nd IEEE Conference on Computer Communications*, 2013.

[6] M. Feily, A. Shahrestani, and S. Ramadass, "A Survey of Botnet and Botnet Detection." in *SECURWARE*, R. Falk, W. Goudalo, E. Y. Chen, R. Savola, and M. Popescu, Eds., IEEE. IEEE Computer Society, 2009, p. 268–273.

[7] A. Ferreiro, T. Fichtel, J. López de Vergara, P. Mátray, F. Strohmeier, G. Tropea, and U. Weinsberg, "Semantic Unified Access to Traffic Measurement Systems for Internet Monitoring Service," *ICTMobileSummit2009, Santander (Spain)*, 2009.

[8] A. Hanemann, J. W. Boote, E. L. Boyd, J. Durand, L. Kudarimoti, R. Lapacz, D. M. Swany, S. Trocha, and J. Zurawski, "PerfSONAR: A Service Oriented Architecture for Multi-domain Network Monitoring." in *ICSOC*, ser. LNSC, B. Benatallah, F. Casati, and P. Traverso, Eds., vol. 3826. Springer, 2005, p. 241–254.

[9] F. Leder and T. Werner, "Know Your Enemy: Containing Conficker. To Tame a Malware," The Honeynet Project, Tech. Rep., April 2009.

[10] J. E. López de Vergara and J. Aracil, "Measurements and Measurement Tools in OpenLab: Use Cases with Measurement Data Ontologies." in *FP7 FIRE/EULER*, ser. LNSC, L. Fàbrega, P. Vilà, D. Careglio, and D. Papadimitriou, Eds., vol. 7586. Springer, 2012, p. 159–174.

[11] P. Mockapetris, "RFC 1035 Domain Names - Implementation and Specification," Internet Engineering Task Force, November 1987.

[12] MOI ETSI ISG, "Measurement Ontology for IP traffic (MOI); Requirements for IP traffic measurement ontologies development," ETSI, Tech. Rep., 07 2012.

[13] ——, "Measurement Ontology for IP traffic (MOI); IP traffic measurement ontologies architecture. DGS/MOI-003 ," ETSI, Tech. Rep., 05 2013.

[14] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford University, Tech. Rep., 2001.

[15] E. Prud'hommeaux, C. Buil-Aranda *et al.*, "SPARQL 1.1 Federated Query," W3C, Mar. 2013, http://www.w3.org/TR/sparql11-federated-query.

[16] S. S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. T. Jr, S. Auer, J. Sequeda, and A. Ezzat, "A Survey of Current Approaches for Mapping of Relational Databases to RDF." W3C, W3C RDB2RDF Incubator Group, Tech. Rep., January 2009.

[17] A. Salvador, J. E. López De Vergara, G. Tropea, N. Blefari-Melazzi, A. Ferreiro, and A. Katsu, "A Semantically Distributed Approach to Map IP Traffic Measurements to a Standardized Ontology," *IJCNC*, vol. 2, no. 1, 2010.

[18] A. Wong, P. Ray, N. Parameswaran, and J. Strassner, "Ontology mapping for the interoperability problem in network management," *Selected Areas in Communications, IEEE Journal on*, vol. 23, no. 10, p. 2058–2068, 2005.

[19] H. Xu and D. Xiao, "A Common Ontology-Based Intelligent Configuration Management Model for IP Network Devices," in *Innovative Computing, Information and Control*, 2006, p. 385–388.

[20] J. Zurawski, D. M. Swany, and D. Gunter, "A scalable framework for representation and exchange of network measurements." in *TRIDENTCOM*. IEEE, 2006.